IN THE CLAIMS

Please amend the claims as follows.

For the Examiner's convenience, a list of all claims is included below.

1. (Currently Amended) A method comprising:

receiving a binary of a program code, the binary based on a first instruction set architecture;~~ and~~

translating the binary, wherein the translated binary is based on a combination of the first instruction set architecture and a second instruction set architecture~~.~~; and

executing the translated binary.

2. (Currently Amended) ~~The~~A method ~~of claim 1~~comprising:

receiving a binary of a program code, the binary based on a first instruction set architecture;

translating the binary, wherein the translated binary is based on a combination of the first instruction set architecture and a second instruction set architecture; and

executing the translated binary,

further comprising checking instruction set architecture execution flags, the instruction set architecture execution flags to indicate at least one translation of a portion of the binary.

3. (Original) The method of claim 2, wherein the instruction set architecture execution flags are set by a programming environment of the binary.

4. (Original) The method of claim 2, wherein a register in a processor translating the binary is to store the instruction set architecture execution flags.

5. (Cancelled)

6. (Original) The method of claim 5, wherein the translating and executing are based on a command, the instruction set architecture execution flags based on a number of command line flags associated with the command.

7. (Original) The method of claim 1, wherein the first instruction set architecture comprises floating-point instructions and wherein the second instruction set architecture comprises floating-point instructions, wherein the translating of the binary comprises translating the floating-point instructions of the first instruction set architecture to the floating-point instructions of the second instruction set architecture.

8. (Original) The method of claim 1, wherein the translating of the binary comprises storing a portion of a hardware stack in a register of a processor translating the binary.

9. (Currently Amended) A method comprising:

receiving a binary of a program code, the binary based on a first instruction set architecture; and

executing the binary, wherein the executing comprises translating at least one instruction of the binary based on the first instruction set architecture to at least one instruction based on a second instruction set architecture, further comprising checking instruction set architecture execution flags, the instruction set architecture execution flags to indicate at least one translation of a portion of the binary, the translated binary based on a combination of first instruction set architecture and the second instruction set architecture.

10. (Original) The method of claim 9, wherein the first instruction set architecture includes in-order accesses to memory and the second instruction set architecture includes out-of-order accesses to memory, the translating of the binary to include out-of-order accesses to memory by a processor executing the binary.

11. (Original) The method of claim 9, wherein the first instruction set architecture allows the binary to self modify and the second instruction set architecture does not allow the binary to self modify, the translating of the binary to include an instruction to controllers of memories that store the binary to perform write operations independent of checks of whether the write operations modify a location where the binary is stored.

12. (Original) The method of claim 9, wherein the second instruction set architecture has an address space that is larger than the first instruction set architecture, translating of the binary comprises using the address space of the second instruction set architecture.

13. (Original) The method of claim 12, wherein data accessed by the binary is stored in a single segment in memory and wherein an offset value for translating a virtual address to a physical address for the data is not modified during execution of the binary.

14. (Original) A system comprising:

a memory to include a binary of a program code based on a first instruction set architecture; and

a processor coupled to the memory, the processor to execute the binary, wherein executing the binary comprises translating the binary, the translated binary based on a combination of the first instruction set architecture and a second instruction set architecture.

15. (Original) The system of claim 14, wherein the processor comprises a register to store instruction set architecture execution flags, the instruction set architecture execution flags to indicate at least one translation of a portion of the binary.

16. (Original) The system of claim 15, wherein the instruction set architecture execution flags are set by a programming environment of the binary.

17. (Original) The system of claim 14, wherein the second instruction set architecture has an address space that is larger than the first instruction set architecture, the translating of the binary comprises using the address space of the second instruction set architecture.

18. (Original) The system of claim 17, wherein the binary is stored in a single segment in the memory and wherein an offset value for translating a virtual address to a physical address is not modified during execution of the binary.

19. (Original) An apparatus comprising:

a decoder to receive a binary based on a first instruction set architecture; and

a number of registers, wherein at least one of the number of registers is to store instruction set architecture execution flags, the instruction set architecture execution flags to indicate a translation of a binary, the translated binary based on a combination of the first instruction set architecture and a second instructions set architecture.

20. (Original) The apparatus of claim 19, wherein the first instruction set architecture comprises floating-point instructions and wherein the second instruction set architecture comprises floating-point instructions, wherein the translating of the binary comprises translating the floating-point instructions of the first instruction set architecture to the floating-point instructions of the second instruction set architecture.

21. (Original) The apparatus of claim 19, wherein the translating of the binary comprises storing a portion of a hardware stack in a register within the number of registers.

22. (Original) The apparatus of claim 19, wherein the apparatus is coupled to memories to store the binary, wherein the first instruction set architecture allows the binary to self modify and the second instruction set architecture does not allow the binary to self modify, the translating of the binary to include an instruction to controllers of the memories to perform write operations independent of checks of whether the write operations modify a location where the binary is stored.

23. (Original) The apparatus of claim 19, wherein the second instruction set architecture has an address space that is larger than the first instruction set architecture, the translating of the binary comprises using the address space of the second instruction set architecture.

24. (Original) The apparatus of claim 23, wherein data accessed by the binary is stored in a single segment in memory coupled to the apparatus and wherein an offset value for translating a virtual address to a physical address for the data is not modified during execution of the binary.

25. (Currently Amended) A machine-readable medium that provides instructions, which when executed by a machine, causes the machine to perform operations comprising:

receiving a binary of a program code, the binary based on a first instruction set architecture;

translating the binary, wherein the translated binary is based on a combination of the first instruction set architecture and a second instruction set architecture; and

executing the binary.

26. (Cancelled)

27. (Original) The machine-readable medium of claim ~~26~~25, wherein the translating and executing are based on a command, ~~the~~wherein instruction set architecture execution flags based on a number of command line flags associated with the command.

28. (Original) The machine-readable medium of claim 25, wherein the first instruction set architecture comprises floating-point instructions and wherein the second instruction set architecture comprises floating-point instructions, wherein the translating of the binary comprises translating the floating point instructions of the first instruction set architecture to the floating-point instructions of the second instruction set architecture.

29. (Original) The machine-readable medium of claim 25, wherein the first instruction set architecture allows the binary to self modify and the second instruction set architecture does not allow the binary to self modify, the translating of the binary to include an instruction to controllers of memories that store the binary to perform write operations independent of checks of whether the write operations modify a location where the binary is stored.